

Fingerprint Security System using GSM and GPRS

Dr. V. P. Patil, Preeti Bhosale, Suraj Mane, Nishant Parsekar, Pranav Sarvankar

Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India

ABSTRACT

The objective of this project is to create and implement a security system based on fingerprint recognition and GSM/GPRS technology that can be utilized in various settings, such as offices, homes, and banks. The system ensures that only authorized individuals are granted access to secure entry and exit points. Fingerprint recognition and GSM/GPRS technology are utilized in this security system, enabling the lock to open only if the individual scanning their fingerprint is authenticated. Furthermore, when a fingerprint is scanned, the system captures an image of the person, which is then sent to a registered email address. Additionally, a message is sent to a registered mobile number using GSM technology. Email access is facilitated by GPRS technology. In the event that an unauthorized individual attempts to scan their fingerprint, a buzzer will sound, alerting the authorities that an unknown person is attempting to gain access. A photograph of the individual is then captured and sent to the registered email address, and a message is sent to the registered mobile number. Biometric technology and the security system offer numerous benefits over traditional systems. This system can generate a log that records the check-in and check-out times of each user, as well as basic information.

How to cite this paper: Dr. V. P. Patil | Preeti Bhosale | Suraj Mane | Nishant Parsekar | Pranav Sarvankar "Fingerprint Security System using GSM and GPRS" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-7 | Issue-2, April 2023, pp.867-876, www.ijtsrd.com/papers/ijtsrd56188.pdf URL:



Copyright © 2023 by author (s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



INTRODUCTION

In today's society, people are becoming increasingly concerned about the security of their valuable belongings, such as jewellery, money, and important documents. As a result, bank lockers are commonly used as a secure storage solution. However, with the advent of fast-evolving technologies, users are looking for high-security systems that incorporate electronic identification options. These identification technologies include bank lockers, ATMs, intelligent cards, user IDs, password-based systems, and more. Unfortunately, these systems are not always protected from attacks by hackers, theft, and forgotten passwords.

Global System for Mobile communication (GSM) is a communication technology that is primarily used for sending or receiving data, such as text messages. In our security system, GSM plays an important role. Through the use of GSM, the user is immediately alerted via a text message if an unauthorized person attempts to open the locker. Our project utilizes fingerprint, password, and GSM technology-based security systems to provide a more efficient and reliable security system than traditional methods. Additionally, General Packet Radio Service (GPRS)

is used to send emails to registered email addresses. In conjunction with GSM, GPRS plays a major role in our project by sending an email containing the image of the person who attempted to open the locker, whether authorized or not. This approach ensures a high level of security and allows for easy tracking of entry and exit via biometrics.

Literature Survey:

Literature survey on fingerprint security system using GSM reveals that this technology is gaining popularity due to its advantages over traditional access control systems. The following are some studies that highlight the potential of this technology:

"Design and Implementation of a Fingerprint-Based Security System for ATM Using GSM and GPRS" by Oyewole and Adebisi (2017) presents a fingerprint security system that allows users to access an ATM using their fingerprints. The system uses GSM and GPRS to transmit data and control the access to the ATM. The authors report high accuracy in fingerprint recognition and reliable data transmission.

"Fingerprint Access Control System Based on GSM and GPRS" by Gao et al. (2018) presents a fingerprint

access control system that uses GSM and GPRS for remote monitoring and data transmission. The system is designed for use in small to medium-sized buildings and provides secure access control using fingerprint biometrics. The authors report high accuracy in fingerprint recognition and reliable data transmission.

"A Study on Fingerprint Biometric System with GSM/GPRS" by Pudukudy and Soman (2015) presents a study on the integration of fingerprint biometrics with GSM and GPRS technologies for secure access control applications. The authors report high accuracy in fingerprint recognition and reliable data transmission using the proposed system.

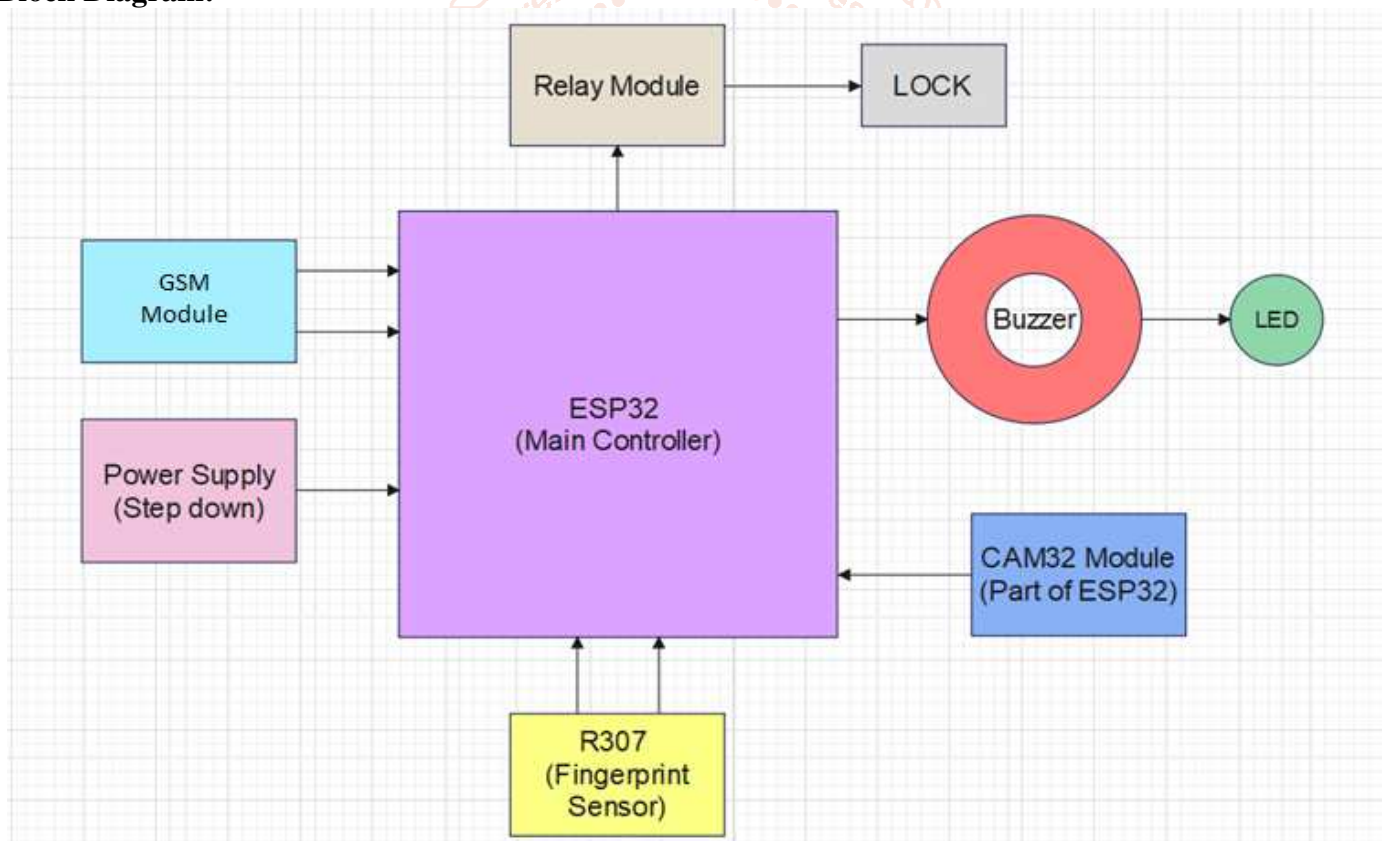
"Design and Implementation of a Fingerprint-Based Security System with GSM Interface" by O. Adeyemo et al.: The study proposed a fingerprint-based security system with a GSM interface that can be used to secure homes, offices, and other small-scale applications. The system was evaluated in terms

of its reliability and effectiveness in providing access control.

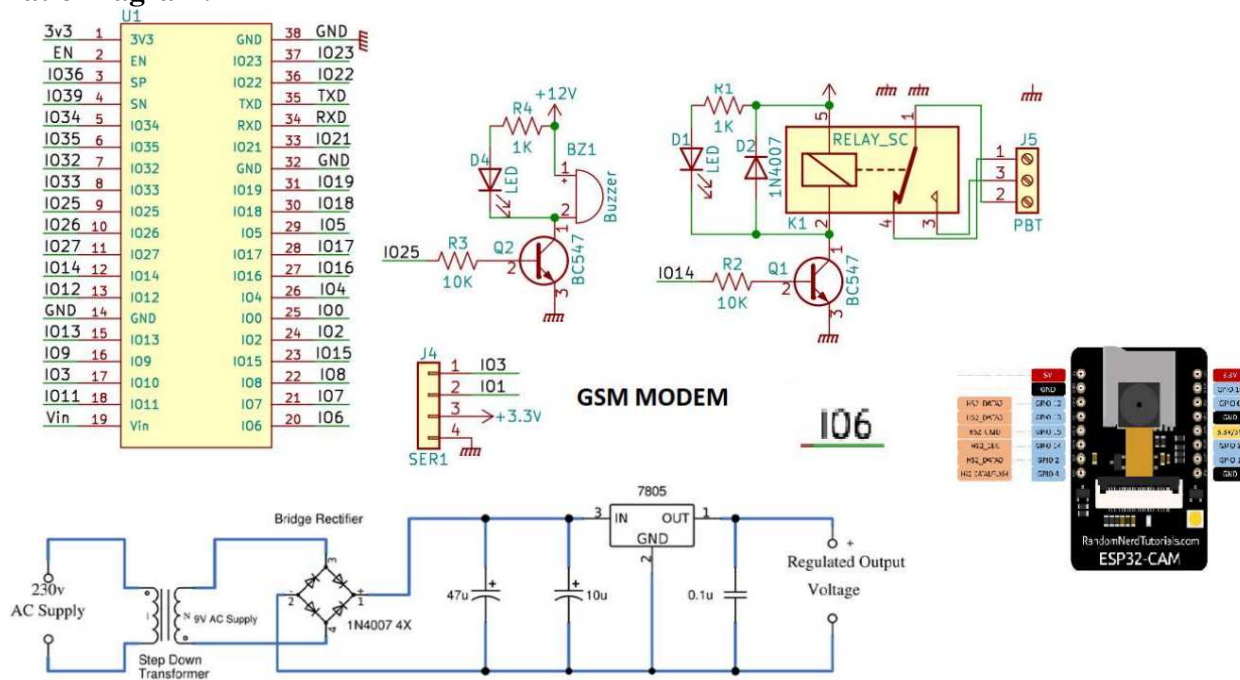
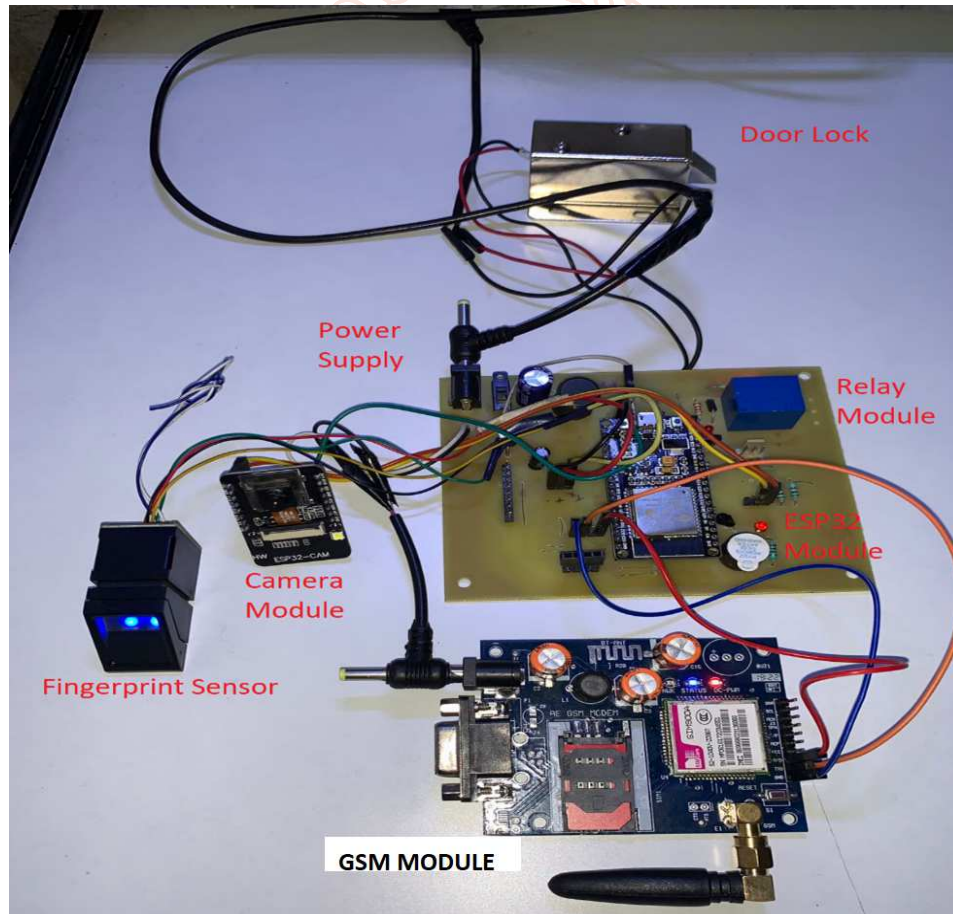
"Fingerprint Identification System Based on GSM and GPRS Technology" by Zhang and Hu (2019) presents a fingerprint identification system that uses GSM and GPRS technology for data transmission and remote monitoring. The system is designed for use in secure access control applications, such as building entrances, vaults, and laboratories. The authors report high accuracy in fingerprint recognition and reliable data transmission.

"Development of a Fingerprint Access Control System Using GSM and GPRS Technologies" by Abdulqadir and Abdulqadir (2019) presents a fingerprint access control system that uses GSM and GPRS technologies for data transmission and remote monitoring. The system is designed for use in small to medium-sized buildings and provides secure access control using fingerprint biometrics. The authors report high accuracy in fingerprint recognition and reliable data transmission.

Block Diagram:



ESP32 is used as a main controller in this system. Internal peripherals of the ESP32 are configured for the system to work. WiFi module is connected to the ESP32 via serial port 1 and the Fingerprint sensor is connected to the serial port 2. Buzzer is connected to ESP32 via a GPIO and LED indication is connected parallel to the buzzer. Hence whenever buzzer will blow LED will also indicate the same. Relay module is connected to ESP32 via a GPIO. Whenever relay will switch then automatically lock will open and close with respect to the state of the relay. CAM32 module is also connected to the ESP32 via a GPIO in accordance with the controlling pin. WiFi module provides GSM and GPRS interfacing. Power supply gives the power to the ESP32 for functioning. Power supply is having step down transformer and 7805 regulator to convert the 12V output of the transformer to the regulated 5V.

Schematic Diagram:**Actual Project:****Working:**

The system is totally built up on the ESP32 module. General purpose Input and Output port of the ESP32 is used to handle Relay, Buzzer, Camera module. Wi-Fi Module and the Fingerprint sensor is access using serial ports.

A fingerprint security system is a biometric authentication method that uses a person's unique fingerprint to grant access to a secured location or

device. The system works by capturing an image of the person's fingerprint and analyzing it to determine if it matches the stored template of an authorized user. Here's how a fingerprint security system typically works:

Initialisation: At Power ON, power supply turns on the system and initialised all the peripherals. GSM module, GPRS Module, CAM32 and ESP32.

Enrolment: The user's fingerprint is captured and stored in the system's database. The fingerprint is usually scanned several times from different angles to create a complete and accurate image.

Authentication: When the user wants to gain access to a secured location or device, they place their finger on a fingerprint scanner. The scanner captures an image of their fingerprint and compares it to the stored template in the database.

Verification: If the captured fingerprint matches the stored template, the system grants access to the user and GSM module also send a message to the registered mobile number that that "Access given to unlock the door". If captured fingerprint does not match then access is denied. Buzzer blows and LED indication turn ON to indicate that un-authorised person tried to access the door. Also when buzzer turned ON, camera module turns ON and image of the person get captured. As soon as image is captured GPRS module is enabled and image is sent to the registered Emil ID. Also GSM module send message to the registered mobile number that "Un-authorised person tried to access the door".

The fingerprint security system works by using advanced algorithms to analyze the unique features of a person's fingerprint, such as the ridges, valleys, and minutiae points. These features are used to create a mathematical representation of the fingerprint, which is then compared to the stored template for authentication.

Fingerprint security systems are highly secure and accurate, as each person's fingerprint is unique and cannot be replicated. Additionally, they are easy to use and convenient, as users do not need to remember passwords or carry access cards.

Program:

```
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>

int BUZZ = 25;
int RELAY1 = 14;

//*****ProjectDefine*****//

#define COUNTRY_CODE "+91"
#define MOBILE_NO_1 "8652882630"

#define TIME_OF_RESPONSE 1000
#define DOOR_OPEN_DELAY 6000
#define SEND_PHOTO_DELAY 5000

String MobileNo1 = String(COUNTRY_CODE) +
String(MOBILE_NO_1);
int fing Action Counter = 0;

//*****//
```

```
//Create all require OBJCETS
```

```
//Finger PRINT Objects creation
```

```
#if (defined(__AVR__) || defined(ESP8266)) &&
!defined(__AVR_ATmega2560__)
```

```
// For UNO and others without hardware serial, we
must use software serial...
```

```
// pin #2 is IN from sensor (GREEN wire)
```

```
// pin #3 is OUT from arduino (WHITE wire)
```

```
// Set up the serial port to use softwareserial..
```

```
SoftwareSerial mySerial(2, 3);
```

```
#else
```

```
// On Leonardo/M0/etc, others with hardware serial,
use hardware serial!
```

```
// #0 is green wire, #1 is white
```

```
#define mySerial Serial2
```

```
#endif
```

```
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
```

```
// The serial object GSM module
```

```
//SoftwareSerial GSM_Serial(GSM_RX, GSM_TX);
```

```
#define GSM_Serial Serial1
```

```
//*****//
```

```
void setup()
```

```
{
Serial.begin(ESP32_BAUDRATE);
```

```
GSM_Serial.begin(GSM_BAUDRATE);
```

```
pinMode (RELAY1,OUTPUT);
```

```
pinMode (BUZZ ,OUTPUT);
```

```
//configure pin 2 as an input and enable the internal
pull-up resistor
```

```
pinMode(CAPTURE_IMAGE_PIN, OUTPUT); //
```

```
Applying ZERO volt will toggle
```

```
offRelay(CAPTURE_IMAGE_PIN);
```

```
printOnLCD("Initializing...");
```

```
initFingerPrint();
```

```
gsmInit();
```

```
printOnLCD("Initializing Done.");
```

```
delay(2000);
```

```
digital Write(RELAY1,HIGH);
```

```
delay(5000);
```

```
digital Write(RELAY1,LOW);
```

```
}
```

```
void loop()
```

```
{
```

```
digital Write(RELAY1,LOW);
```

```
uint8_t status FP = getFingerprintID();
```

```
switch(status FP)
```

```
{
```

```
case FINGERPRINT_OK:
```

```
//OK status
```

```
printOnLCD("Sending SMS...");
```

```

sendSMS(MobileNo1, "Access given to unlock Door.");
digital Write(RELAY1,HIGH);
delay(5000);
digital Write(RELAY1,LOW);
printOnLCD("SMS Sent.");
fing Action Counter = 0;
delay(2000);
break;
case FINGERPRINT_NOTFOUND:
fing Action Counter = fing Action Counter + 1;
if(fing Action Counter >= 2)
{
digital Write(BUZZ,HIGH);
printOnLCD("Sending SMS...");
sendSMS(MobileNo1, "Un-authorised person tried to access Door");
printOnLCD("Sending Photo in Email...");
sendPhoto();
delay(5000);
printOnLCD("Photo Sent.");
delay(1000);
fing Action Counter = 0;
digital Write(BUZZ,LOW);
}
break;
default:
//Error
break;
}
delay(2000);
}
//*****//

void initFingerPrint()
{
Serial.println("\n\nFingerprint Sensor Detect test...");

// set the data rate for the sensor serial port
finger.begin(FINGER_PRINT_BAUDRATE);
delay(5);
if (finger.verifyPassword())
{
Serial.println("Found fingerprint sensor!");
}
else
{
Serial.println("Did not find fingerprint sensor :(");
while (1) { delay(1); }
}

finger.getTemplateCount();

if (finger.templateCount == 0)
{
Serial.print("Sensor doesn't contain any fingerprint data. Please run the 'enroll' example.");
}
}

else
{
Serial.println("Waiting for valid finger...");
Serial.print("Sensor contains ");
Serial.print(finger.templateCount);
Serial.println(" templates");
}
}

bool readFinger()
{
uint8_t status FP = getFingerprintID();

switch(status FP)
{
case FINGERPRINT_OK:
//OK status
return (true);
break;
case FINGERPRINT_NOTFOUND:
default:
//Error
return (false);
}
return (true);
}

uint8_t getFingerprintID()
{
uint8_t p = finger.getImage();
switch(p)
{
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println("No finger detected");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK success!
p = finger.image2Tz();
switch(p)
{

```

```

case FINGERPRINT_OK:
Serial.println("Image converted");
break;

case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;

case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;

case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;

case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;

default:
Serial.println("Unknown error");
return p;
}

// OK converted!

p = finger.fingerSearch();
if (p == FINGERPRINT_OK)
{
Serial.println("Found a print match!");
}
else if (p == FINGERPRINT_PACKETRECEIVEERR)
{
Serial.println("Communication error");
return p;
}
else if (p == FINGERPRINT_NOTFOUND)
{
Serial.println("Did not find a match");
return p;
}
else
{
Serial.println("Unknown error");
return p;
}

// found a match!
Serial.print("Found ID #");
Serial.print(finger.fingerID);
Serial.print(" with confidence of ");
Serial.println(finger.confidence);

//return finger.fingerID;
return FINGERPRINT_OK;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprntIDez()
{
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;

p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;

p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;

// found a match!
Serial.print("Found ID #");
Serial.print(finger.fingerID);
Serial.print(" with confidence of ");
Serial.println(finger.confidence);
return finger.fingerID;
}

void onRelay(int pin)
{
if( RELAY_LOG == HIGH )
{
digital Write(pin, HIGH);
}
else
{
digital Write(pin, LOW);
}
}

void offRelay(int pin)
{
if( RELAY_LOG == HIGH )
{
digital Write(pin, LOW);
}
else
{
digital Write(pin, HIGH);
}
}

void onLed(int pin)
{
if( LED_LOG == HIGH )
{
digital Write(pin, HIGH);
}
else
{
digital Write(pin, LOW);
}
}

void offLed(int pin)
{
if( LED_LOG == HIGH )
{
digital Write(pin, LOW);
}
}

```

```

}
else
{
digital Write(pin, HIGH);
}
}

void gsmInit()
{
//Deleting Messages
for(int i = 1; i < 10 ;i++)
{
Serial.println(sendCommandAndGetResponse("AT+
CMGD=" + String(i) ));
}
}

void sendSMS(String mob_no, String msg)
{
Serial.println(sendCommandAndGetResponse("AT+
CMGS=\"" + mob_no + "\""));
Serial.println(sendCommandAndGetResponse(msg));
}

String      sendCommandAndGetResponse(String
command)
{
long prev_time;
//Serial output i arudio terminal
printOnLCD(command);
GSM_Serial.println(command);

prev_time = millis();

while( (    millis()    -    prev_time    )
TIME_OF_RESPONSE )
{
if( GSM_Serial.available() > 0)
{
return GSM_Serial.readString();
}
}

return "";
}

int checkForNewSMS()
{
int msg_no;
//printOnLCD("Waiting for any new message...");
while(true)
{
if( GSM_Serial.available() > 0 )
{
String str = GSM_Serial.readString();
if( str.indexOf("+CMTI: \"SM\",") > 0 )
{
str.trim();
if( str.length() > 13 )
{
msg_no = (str.substring(12, 14)).toInt();
}
else
{
msg_no = (str.substring(12, 13)).toInt();
}
return msg_no;
}
else
{
return -1;
}
}
}

//This function will only returns new message if
receive. It does not check for any mobile number
String getActualMessage(int mesg_no)
{
String      str      =
sendCommandAndGetResponse("AT+CMGR=" +
String(mesg_no) );
//printOnLCD("Mess Contents:\n\n" + str);
for(int i = 20; i < str.length() ; i++)
{
if( str.charAt(i) == '\n' )
{
return str.substring((i+1), (str.length()-8) );
}
}

//This function will only return message value for
correct mobile no.
String getActualMessage(int  mesg_no, String
from_no)
{
int len = 60;
String      str      =
sendCommandAndGetResponse("AT+CMGR=" +
String(mesg_no) );
if( ( str.substring(0, len) ).indexOf(from_no) > 0 )
{
for(int i = 20; i < str.length() ; i++)
{
if( str.charAt(i) == '\n' )
{
return str.substring((i+1), (str.length()-8) );
}
}
}
else
{
return "";
}
}

```

```

}
}

void printOnLCD(String val)
{
if( SERIAL_PRINT_WITH_LCD == true )
Serial.println( "\n" + val );
}

void sendPhoto()
{
onRelay(CAPTURE_IMAGE_PIN);
delay(SEND_PHOTO_DELAY);
offRelay(CAPTURE_IMAGE_PIN);
}

```

Code Explanation:

1. void loop function: The loop function starts by turning off a relay using digital Write function. It then calls a function named getFingerprintID to get the status of the fingerprint sensor, which is stored in a variable named status FP. The program then uses a switch-case statement to check the value of status FP and perform different actions accordingly. If status FP is FINGERPRINT OK, the program prints a message on an LCD display, sends an SMS to a mobile number, turns on the relay for 5 seconds using digital Write function, and resets a counter variable named fing Action Counter to 0. If status FP is FINGERPRINT NOTFOUND, the program increments the value of fing Action Counter by 1. If fing Action Counter becomes greater than or equal to 2, the program sounds a buzzer using digital Write function, sends an SMS to a mobile number, sends a photo in an email, and resets fing Action Counter to 0. If status FP is any other value, the program does nothing. Finally, the program waits for 2 seconds using the delay function and repeats the loop.
2. void initFingerPrint function: This code defines a function named initFingerPrint that initializes the fingerprint sensor and performs some basic checks to ensure that it is working properly. The function starts by printing a message on the Serial monitor to indicate that the fingerprint sensor is being detected. It then initializes the sensor by setting the data rate for the sensor serial port and verifying its password using the verify Password function of the Adafruit Fingerprint library. If the password is verified, the program prints a message indicating that the sensor has been found. If not, the program prints an error message and enters an infinite loop. The program then calls the getTemplateCount function to get the number of templates stored in the sensor. If the sensor does not contain any templates, the

program prints an error message and suggests running the enroll example. Otherwise, the program prints a message indicating that it is waiting for a valid finger and the number of templates stored in the sensor.

3. bool read Finger function: This code defines a function named read Finger that reads the fingerprint sensor and returns a boolean value based on the status of the sensor. The function calls the get Fingerprint ID function to get the status of the fingerprint sensor, which is stored in a variable named status FP. The program then uses a switch-case statement to check the value of status FP and perform different actions accordingly. If status FP is FINGERPRINT OK, the function returns true, indicating that the fingerprint has been recognized. If status FP is FINGERPRINT NOTFOUND or any other value, the function returns false, indicating that the fingerprint has not been recognized.

4. uint8 t getFingerprintID function:

The function getFingerprintID() retrieves the ID of a fingerprint from the fingerprint sensor. Firstly, it calls the getImage() function to capture an image of the fingerprint, and the status of the fingerprint sensor is stored in a variable named 'p'. The function uses a switch-case statement to check the value of 'p' and perform different actions accordingly. If 'p' is FINGERPRINT_OK, the function prints "Image taken" the program then outputs the results to the serial monitor and continues to the next step. If 'p' is FINGERPRINT_NOFINGER, the function prints "No finger detected" to the serial monitor and returns 'p'. If 'p' is FINGERPRINT_PACKETRECEIVEERR, the function prints "Communication error" to the serial monitor and returns 'p'. If 'p' is FINGERPRINT_IMAGEFAIL, the function prints "Imaging error" to the serial monitor and returns 'p'. If 'p' is any other value, the function prints "Unknown error" to the serial monitor and returns 'p'. The function then calls the image2Tz() function to convert the fingerprint image to a template, and the status of the fingerprint sensor is stored in the variable 'p' again. The function uses another switch-case statement to check the value of 'p' and perform different actions accordingly. If 'p' is FINGERPRINT_OK, the function prints "Image converted" the program then outputs the results to the serial monitor and continues to the next step. If 'p' is FINGERPRINT_IMAGEMESS, the function prints "Image too messy" to the serial monitor and returns p. If p is FINGERPRINT_PACKETRECEIVEERR, the function prints "Communication error" to the serial

monitor and returns p. If p is FINGERPRINT_FEATUREFAIL, the function prints "Could not find fingerprint features" to the serial monitor and returns p. If p is FINGERPRINT_INVALIDIMAGE, the function prints "Invalid image" to the serial monitor and returns p. If p is any other value, the function prints "Unknown error" to the serial monitor and returns p. The function then calls the fingerSearch() function to search for a matching fingerprint template. The status of the fingerprint sensor is stored in the variable p again. The function uses an if-else statement to check the value of p and perform different actions accordingly. If p is FINGERPRINT_OK, the function prints "Found a print match!" the program then outputs the results to the serial monitor and continues to the next step. If p is FINGERPRINT_PACKETRECEIVEERR, the function prints "Communication error" to the serial monitor and returns p. If p is FINGERPRINT_NOTFOUND, the function prints "Did not find a match" to the serial monitor and returns p. If p is any other value, the function prints "Unknown error" to the serial monitor and returns p. If a match is found, the function prints the ID and confidence level of the matching fingerprint to the serial monitor and returns FINGERPRINT_OK. Otherwise, it returns the value of p.

5. intgetFingerprintIDez function:

This is a modified version of the get Fingerprint ID function. Instead of searching the whole database, it uses finger Fast Search function to quickly search for a match against the enrolled fingerprints. This function returns the finger ID of the matched fingerprint or -1 if no match is found.

6. void onRelay, void offRelay, void onLed, void offLed:

These are four functions that control the relay and LED outputs based on the RELAY LOG and LED LOG variables. The on Relay function turns on the relay by setting the specified pin to HIGH if RELAY LOG is HIGH, and LOW if RELAY LOG is LOW. The off Relay function turns off the relay by setting the specified pin to LOW if RELAY LOG is HIGH, and HIGH if RELAY LOG is LOW. The on Led function turns on the LED by setting the specified pin to HIGH if LED LOG is HIGH, and LOW if LED LOG is LOW. The off Led function turns off the LED by setting the specified pin to LOW if LED LOG is HIGH, and HIGH if LED LOG is LOW. The RELAY LOG and LED LOG variables likely determine the polarity of the relay and LED, respectively.

7. The code provided is a set of functions for a GSM module. The functions perform the following tasks:

gsmInit function: Deletes any SMS message stored in the GSM modules memory.

sendSMS String mob no, String msg : Sends an SMS message to the specified mobile number.

sendCommandAndGetResponse String command : Sends a command to the GSM module and returns the response received from the module.

checkForNewSMS function: Checks if there is any new SMS message received by the GSM module and returns the message number.

getActualMessage int mesg no : Gets the actual message from the specified message number.

getActualMessage int mesg no, String from no : Gets the actual message from the specified message number, but only if it is received from the specified mobile number.

printOnLCD String val : Prints the specified string on the serial monitor, if enabled.

sendPhoto function: Sends a signal to capture a photo by activating a relay for a specified delay, after which the relay is deactivated.

Future Scope:

1. Improving fingerprint recognition algorithms: The accuracy and reliability of fingerprint recognition algorithms are essential for any fingerprint security system. Researchers can work on improving the accuracy and reliability of these algorithms to make them more robust and effective.
2. Enhancing communication channel security: GSM and GPRS networks are susceptible to different types of attacks such as interception and eavesdropping. Future research can focus on developing methods to enhance the security of communication channels between the fingerprint scanner and the central server, including using encryption or other secure communication protocols.
3. Integrating other biometric modalities: Fingerprint recognition is just one of many biometric modalities that can be used for security purposes. Future research can explore the feasibility of integrating other biometric modalities such as facial recognition, iris recognition, or voice recognition to enhance the security of the system.
4. Investigating the potential use of blockchain technology: Blockchain technology has gained popularity recently because of its security and decentralization features. Future work can

investigate the feasibility of using blockchain technology to enhance the security and privacy of fingerprint security systems.

5. Evaluating the system's performance in real-world scenarios: While fingerprint security systems using GSM and GPRS show promise in laboratory environments, it is essential to evaluate their performance in real-world scenarios. Future work can focus on conducting large-scale field trials to assess the system's effectiveness and identify potential limitations or challenges.

Conclusion:

In summary, a fingerprint security system that employs GSM and GPRS can serve as a secure and efficient method for authentication and access control. The system's ability to validate a person's identity based on their unique fingerprints makes it difficult for unauthorized individuals to access sensitive information or restricted areas. Additionally, the real-time communication between the fingerprint scanner and the central server facilitated by GSM and GPRS technology enables quick response times and remote monitoring.

Nevertheless, to guarantee the system's effectiveness and security, further research is necessary to improve the accuracy and reliability of fingerprint recognition algorithms, enhance the security of communication channels, explore the integration of other biometric

modalities, investigate the potential for using blockchain technology, and evaluate the system's performance in real-world scenarios.

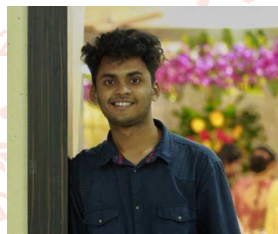
Reference:

- [1] "Fingerprint-Based Security System Using GSM Technology." by T. Manikandan and S. Sivakumar, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2016.
- [2] "Design of Fingerprint-Based Security System Using GSM Technology." by B. V. Venkata Krishna and P. Ravi Kumar, International Journal of Computer Science and Mobile Computing, Vol. 5, Issue 8, August 2016.
- [3] "Fingerprint Recognition System for Access Control and Security Using GSM and GPRS." by D. D. Dhanwate and A. R. Chaudhari, International Journal of Computer Applications, Vol. 95, No. 13, June 2014.
- [4] "Fingerprint Based Security System Using GSM and GPRS." by A. R. Khan and S. A. Khan, International Journal of Computer Applications, Vol. 46, No. 6, May 2012.
- [5] "Design and Implementation of Fingerprint-Based Security System Using GSM and GPRS." by A. R. Khan and S. A. Khan, International Journal of Computer Applications, Vol. 32, No. 9, October 2011.

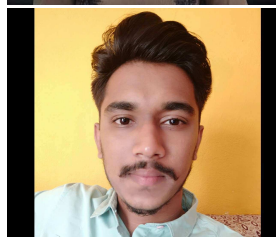
MEET THE TEAM



PREETI BHOSALE
preetibhosale89@gmail.com
Navi Mumbai,
Maharashtra



NISHANT PARSEKAR
nishparsekar@icloud.com
Thane, Maharashtra



SURAJ MANE
surajm161096@gmail.com
Mumbai, Maharashtra



PRANAV SARVANKAR
pranavsarvankar2715@gmail.com
Dombivli, Maharashtra